

Modèles de Calcul : TD3 NP

Exercice 3.1

Justifiez que les réductions polynomiales sont transitives.

Exercice 3.2

Soit la formule booléenne $\varphi = (x_1 \wedge \bar{x}_2) \vee (x_2 \wedge x_4)$.

Expliquer en détail comment construire une formule $\tilde{\varphi}$ en *forme normale conjonctive* (CNF) et telle que φ est satisfaisable ssi $\tilde{\varphi}$ est satisfaisable. Combien de clauses a la formule $\tilde{\varphi}$ en fonction de la taille de φ ?

Exercice 3.3

On considère le problème PROGRAMMATION ENTIÈRE qui teste l'existence d'une solution à un système d'égalités et d'inégalités linéaires. Si x, y, z sont des noms de variables voici par exemple un tel système :

$$\begin{aligned}x + 2 \times y &= z \\ z - x &\leq 3 \times y\end{aligned}$$

Une solution à ce système est par exemple $x = 1, y = 1$ et $z = 3$. Un système est dit *gardé* s'il contient une inégalité de la forme $0 \leq x \leq a$ (avec a un entier) pour tout variable x existante dans le système. On considère le problème PROGRAMMATION ENTIÈRE suivant :

INPUT : Un système \mathcal{S} gardé d'égalités et inégalités linéaires.

QUESTION : Existe-t-il une solution (en nombres entiers) au système ?

Le but de l'exercice est de montrer que PROGRAMMATION ENTIÈRE est NP-complet.

1. Quel est votre avis, pourquoi restreint-on le problème aux systèmes qui sont gardés ?
2. Justifiez que PROGRAMMATION ENTIÈRE est dans NP.
3. On désire maintenant montrer que PROGRAMMATION ENTIÈRE est NP-difficile. Pour cela on construit une réduction polynomiale de 3-SAT à PROGRAMMATION ENTIÈRE. À toute clause de 3-SAT on va associer une égalité linéaire. On commence par un exemple.
 - (a) Soit $C = p_1 \vee p_2 \vee \neg p_3$ une clause de 3-SAT. Donnez une expression $E(x_1, x_2, x_3)$ telle que $p_1 \vee p_2 \vee \neg p_3$ s'évalue à vrai ssi $E(p_1, p_2, p_3)$ est différent de 0 (où $E(p_1, p_2, p_3)$ s'obtient en substituant x_i par 1 si p_i est vrai et par 0 sinon).
 - (b) Généraliser l'exemple précédent et donner une réduction polynomiale de 3-SAT à PROGRAMMATION ENTIÈRE. En conclure que PROGRAMMATION ENTIÈRE est un problème NP-difficile.
4. Conclure que PROGRAMMATION ENTIÈRE est un problème NP-complet.

Exercice 3.4

On considère maintenant le problème COUVERTURE PAR SOMMETS (VERTEX COVER). Soit $G = (V, E)$ un graphe, un sous-ensemble de sommets $C \subseteq V$ est dit *couvrant* si pour toute arête $(u, v) \in E$ au moins une de ses extrémités u ou v appartient à C . Le problème COUVERTURE PAR SOMMETS est défini ainsi :

INPUT : Un graphe non-orienté $G = (V, E)$ et un entier p .

QUESTION : Existe-t-il un sous-ensemble couvrant $C \subseteq V$ tel que $|C| = p$.

On veut montrer que COUVERTURE PAR SOMMETS est NP-complet.

1. Justifiez que COUVERTURE PAR SOMMETS est dans NP.
2. Il faut maintenant montrer que COUVERTURE PAR SOMMETS est NP-difficile. On va réduire 3-SAT à COUVERTURE PAR SOMMETS. Il faut donc un algorithme polynomial qui, étant donné une instance $\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_m$ de 3-SAT (chaque C_i étant de la forme $\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$), construit un graphe G et un entier p qui satisfont COUVERTURE PAR SOMMETS ssi φ est satisfiable. Dans les questions qui suivent on décrit petit à petit comment construire G . On appelle x_1, \dots, x_n les variables contenues dans les clauses de φ .
 - (a) Soit G un graphe à $2n$ sommets $\{q_1, \dots, q_n, r_1, \dots, r_n\}$ avec arêtes $\{(q_i, r_i) \mid i = 1, \dots, n\}$. Qu'est-ce qu'on peut dire a propos de tout ensemble couvrant minimal de G ?
 - (b) Soit G un graphe avec 3 sommets distingués s_1, s_2, s_3 , tel que la taille minimale d'un ensemble couvrant est n . Compléter G avec 3 nouveaux sommets et quelques arêtes tel que tout ensemble couvrant de taille $n + 1$ du graphe résultant contient au moins deux des trois sommets s_1, s_2, s_3 .
 - (c) Utiliser les questions précédentes pour donner une réduction polynomiale de 3-SAT à COUVERTURE PAR SOMMETS.
3. Conclure que COUVERTURE PAR SOMMETS est NP-complet.

Exercice 3.5

On étudie maintenant le problème CLIQUE :

INPUT : Un graphe non-orienté G et un entier q .

QUESTION : Existe-t-il un sous-graphe *complet* de G ayant q sommets (une q -clique) ?

1. Montrer que COUVERTURE PAR SOMMETS et CLIQUE se réduisent mutuellement l'un à l'autre.
2. Montrer que CLIQUE est NP-complet.

Exercice 3.6

Le problème SOUS-GRAPHE est le suivant :

INPUT : Deux graphes non-orientés G et H .

QUESTION : Existe-t-il un sous-graphe de G isomorphe à H ?

Le problème ISOMORPHISME DE GRAPHE est le suivant :

INPUT : Deux graphes non-orientés G et H .

QUESTION : Ces deux graphes sont-ils isomorphes l'un à l'autre ?

1. Montrer que CLIQUE se réduit à SOUS-GRAPHE.
2. Montrer que SOUS-GRAPHE est NP-complet.
3. Montrer que ISOMORPHISME DE GRAPHES se réduit à SOUS-GRAPHE.

Remarque. Savoir si ISOMORPHISME DE GRAPHES est NP-complet ou non est un problème ouvert. Une conviction générale est que ce problème est de *complexité intermédiaire* : plus compliqué que polynomial mais moins compliqué que NP-complet (tout cela, bien sûr, si $P \neq NP$).

Exercice 3.7

On considère le problème SOMME D'ENTIERS suivant :

INPUT : Des entiers positifs x_1, x_2, \dots, x_n et un entier N .

QUESTION : Existe-t-il une sous-suite $1 \leq i_1 < i_2 < \dots < i_p \leq n$ telle que $x_{i_1} + x_{i_2} + \dots + x_{i_p} = N$?

1. On suppose que l'ensemble $X = \{x_1, \dots, x_n\}$ est déjà trié $x_1 \leq x_2 \leq \dots \leq x_n$. Pour le problème SOMME D'ENTIERS proposer un algorithme qui utilise une mémoire de taille $O(N)$ et dont le temps de calcul est $O(nN)$.
2. Il est connu que le problème SOMME D'ENTIERS est NP-complet. Pourquoi l'existence de l'algorithme de la question 1 ne contredit pas la NP-complétude du problème ?

Exercice 3.8

Un *chemin Hamiltonien* dans un graphe G est un chemin qui passe une et une seule fois par chaque sommet du graphe. On appelle CHEMIN HAMILTONIEN le problème associé :

INPUT : Un graphe orienté G .

QUESTION : Existe-t-il un chemin Hamiltonien dans G ?

Un *circuit Hamiltonien* est un chemin Hamiltonien fermé, c'est-à-dire qu'il existe une arête entre le premier sommet du chemin et le dernier. On appelle CHEMIN HAMILTONIEN le problème suivant :

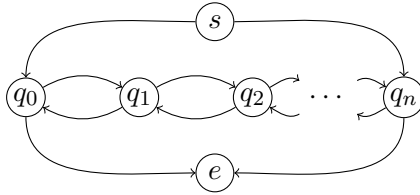
INPUT : Un graphe orienté G .

QUESTION : Existe-t-il un circuit Hamiltonien dans G ?

1. Donner un exemple d'un graphe qui contient un chemin Hamiltonien mais ne contient pas de circuit Hamiltonien.
2. Montrer que CHEMIN HAMILTONIEN se réduit à CIRCUIT HAMILTONIEN.
3. Montrer que CIRCUIT HAMILTONIEN se réduit à CHEMIN HAMILTONIEN.

On cherche maintenant à montrer que CHEMIN HAMILTONIEN est NP-complet. On va réduire 3-SAT, soit \mathcal{C} une instance de 3-SAT, x_1, \dots, x_k les variables et n le nombre de clauses.

1. On considère le graphe G suivant à $n + 3$ sommets :



Montrer qu'il n'existe que deux chemins Hamiltoniens possibles et les donner.

2. En utilisant la question précédente donner un graphe tel que chaque chemin Hamiltonien dans celui-ci correspond à une affectation des variables x_1, \dots, x_k à une valeur de vérité.
3. Complétez le graphe avec de nouveaux sommets et arrêtes pour interdire les chemins Hamiltoniens qui correspondent à des affectations qui ne satisfont pas \mathcal{C} .
4. En déduire que CHEMIN HAMILTONIEN et CIRCUIT HAMILTONIEN sont NP-complets.

Exercice 3.9

Soit $G = (V_G, E_G), H = (V_H, E_H)$ deux graphes, un *morphisme* de G dans H est une application $f : V_G \rightarrow V_H$ telle que :

$$(u, v) \in E_G \implies (f(u), f(v)) \in E_H$$

On considère MORPHISME DE GRAPHES le problème associé :

INPUT : Deux graphes G, H .

QUESTION : Existe-t-il un morphisme de G dans H ?

Prouver que MORPHISME DE GRAPHES est NP-complet. (**Indice** : On pourra réduire 3-COLORABILITÉ)

Exercice 3.10

Soit un graphe orienté $G = (V, E)$ où V est l'ensemble des sommets, et E est l'ensemble des arrêtes du graphe. On dit qu'un sous-ensemble $U \subseteq V$ est le *noyau* du graphe G si :

1. Quels que soient $u_1, u_2 \in U$, il n'y a pas d'arc $u_1 \rightarrow u_2$.
2. Pour tout $v \in V \setminus U$ il existe un $u \in U$ tel qu'il y a un arc $u \rightarrow v$.

On appelle NOYAU le problème associé :

INPUT : Un graphe orienté G .

QUESTION : Existe-t-il un noyau dans G ?

Réduire le problème NOYAU au problème SAT.

Exercice 3.11

Voici une *fausse solution* de l'exercice précédent. À toute paire de sommets $i, j \in V$ on associe une variable booléenne x_{ij} dont la valeur est

$$x_{ij} = \begin{cases} 1 & \text{s'il y a un arc } i \rightarrow j, \\ 0 & \text{sinon.} \end{cases}$$

Alors la première propriété du noyau s'exprime par la formule $\varphi_1 = \bigwedge_{i, j \in U} \neg x_{ij}$. La deuxième propriété s'exprime par la formule $\varphi_2 = \bigwedge_{j \in V \setminus U} \bigvee_{i \in U} x_{ij}$. Le noyau dans le graphe G existe si et seulement si la formule $\varphi = \varphi_1 \wedge \varphi_2$ est satisfaisable.

Critiquer cette "solution".